

SOFTWARE ENGINEERING INSTITUTE CAPABILITY MATURITY MODEL ISSUE PAPER

BACKGROUND

In 1984, the Software Engineering Institute (SEI) was established by the government to address DoD's need for improved software because it was apparent that many software developers did not have a defined and standard process for developing software[1]. In order to provide the government with a tool for gauging how well a contractor's processes are defined, SEI developed the Capability Maturity Model (CMM). The SEI CMM is a five-level model that attempts to quantify a software organization's capability to consistently and predictably produce high-quality software products[1]. "The model is designed so that capabilities at lower stages provide progressively stronger foundations for higher stages. Each development stage or 'maturity level' distinguishes an organization's software process capability." [1] For each maturity level there are associated key process areas (KPs). The KPs identify the requirements for achieving each maturity level; therefore, Level 1 does not include KPs since it is the starting point. Table 1 presents the maturity levels and their associated KPs. "When an organization collectively performs the activities defined by the KPs, it can achieve goals considered important for enhancing process capability." [1] For example, there are six KPA goals that must be attained

Table 1
SEI CMM Levels and KPs

Maturity Level	Rating Definition	SEI CMM Definition[2]	KPs[1]
1.	Initial	The processes are special and mostly undefined. Success depends upon the individual effort.	
2.	Repeatable	Basic project management processes to track cost, schedule and functionality. Tools are in place to repeat success achieved on analogous programs.	Requirements management, Software project planning, Software project tracking and oversight, Software subcontract management, Software quality assurance, Software configuration management
3.	Defined	The software process is organization wide and is employed by both management and engineering. The process is documented, standardized and integrated.	Organization process focus, Organization process definition, Training program, Integrated-software management, Software product engineering, Intergroup coordination, Peer reviews
4.	Managed	The detailed measures of the software process are collected, managed, quantified, understood, and controlled.	Quantitative process management, Software quality management
5.	Optimizing	The software process continuously improves by quantified feedback from the process and testing new and creative ideas and technologies.	Defect prevention, Technology-change management, process-change management

before an organization can receive a Level 2 rating. As sighted in Table 1, one of the Level 2 goals is software project planning, which requires that all individuals and groups associated with the process understand the software estimates and plans and commit to supporting them. In general, Level 2 can be achieved by counting source lines of code (SLOC) developed, counting person-hours expended to develop the SLOC, tracking milestones (calendar dates) and counting the software errors and defects.

The government can obtain information on an organization's software development processes via assessments. Assessments are initiated by the organization to aid in the improvement of its software development practices. The assessment can be conducted by the organization itself or by an independent agency (SEI or SEI-licensed assessment vendor). The assessment provides feedback on the organization's current software development capabilities and trains the organization on ways to improve its capabilities. The following are the six-phases of an assessment:[1]

1. In the selection phase, the organization is identified as an assessment candidate, and the qualified assessing organization conducts an executive-level briefing.
2. In the commitment phase, the organization commits to the full assessment process whereby a senior executive signs an assessment agreement.
3. In the preparation phase, the organization's assessment team receives training, and the on-site assessment process is fully planned. All assessment participants are identified and briefed. The maturity questionnaire is completed at this time by the organization.
4. In the assessment phase, the on-site assessment is typically conducted in a week. The assessment team then meets to formulate preliminary recommendations.
5. In the report phase, the entire assessment team helps prepare the final report and present it to the organization's assessment participants and senior management. The report includes team findings and recommendations for actions.
6. In the assessment follow-up phase, the assessed organization's team, with guidance from the independent assessment organization, formulates an action plan. After approximately 18 months, it is recommended that the organization have a reassessment in order to assess progress and sustain the software process improvement cycle.

An assessment conducted by a SEI-certified organization is typically viewed to be more credible and objective than a self assessment. Currently, there are only a few organizations that have achieved a Level 4 or higher rating. Table 2 presents the results of progressing from one level to the next.

Table 2

Level 5	Productivity/Quality Risk
Level 4	
Level 3	
Level 2	
Level 1	

Even though contractors are reporting benefits as a result of implementing the KPAs, they are not providing supporting quantitative back-up data to substantiate their reports. In light of this, provided

below is a qualitative summary of the lessons learned as reported by several participating organizations. Additionally, a detailed discussion will be provided of the Pros and Cons experienced while participating in the SEI's process improvement approach. Finally, NCCA's recommendation for adjusting an organization's effort based on its current SEI maturity rating will be discussed.

LESSONS LEARNED

Many contractors who have altered their software development processes to incorporate the CMM key process areas have identified several important lessons. Provided below is a list of those lessons as identified by the participating organizations:

1. Management's Commitment

In many organizations, management is reluctant to incorporate new processes because they do not have concrete evidence that the change will save time and money. Without the commitment of management to support the new KPAs, the implementation of the new processes would more likely than not result in failures. "Management buy-in is essential to a successful implementation of the progress assessment instrument and process." [3] "The path to improvement requires investment, risk, time, and the pain of cultural change. Delegation is not strong enough to overcome these roadblocks. Commitment is. Process improvement should be tied to the salary or promotion criteria of senior management." [4]

2. Pride of the Organization

In order for the process improvement to be deemed successful, an organization has to take pride in the implementation of the improvements and the results have to be seen and accepted. "Improvements are one-time achievements, but pride feeds on itself and leads to continuous measurable improvement. When the whole organization buys into the improvement and sees the results unfold, it gains a team esprit de corps and from that, pride." [4]

3. Software Technology Center/Focal Point

Having a centralized software technology center contributes to the improvement of the software process maturity. A software technology center is most effective when the majority of the development, project management, administration, technology development, training, and marketing are housed in one organization [4]. It has been shown that organizations with higher maturity levels also have one centralized software organization [6]. In order for an organization to reap the benefits from the implementation of the process improvements, an organization should have a focal point. "Disintegrated, asynchronous improvement is not only inefficient but also ineffective for solving organization-wide problems. Although there is still the need for cell-level improvement teams, there must also be an organizational focal point to plan, coordinate (integrate), and implement organization-wide process improvements." [4]

IMPACTS OF SEI CMM

There are several advantages and disadvantages of employing the key process areas in the software development process. Below is a list of pros and cons associated with incorporating the key process areas listed in Table 1:

Pros

1. Increased Productivity/Decreased Cost

Contractors have reported an increase in productivity due to the improvement of their software development process. “Raytheon yielded a twofold increase in its productivity and a ratio of 7.7-to-1 return on its improvement expenditures, for a savings of \$4.48 million during 1990 for a \$0.58 million investment.”[5] Various organizations have realized benefits from maturing from one level to the next. Productivities have increased from as little as 2.5 percent to as much as 130 percent[6]. “Published studies of software engineering improvements measured by the CMM indicate significant cost savings or profit return. This implies that software testing and maintenance costs were reduced, since the software better met verification and validation requirements...”[1] Some organizations showed a savings of \$2 million to \$3.4 million in project dollars[6]. Contractors have also experienced a decrease in rework, code problems, and retesting costs[6].

2. Increased Competitiveness

It is generally accepted that higher CMM levels lead to better quality software products and therefore a better company reputation. CMM compliance may also change the manner in which a company interacts with its customers, because there are stringent requirements for maintaining a high maturity level. Highly rated organizations are more adept at handling quick demands by the customer. Fortunately, compliance leads to higher quality software at lower cost. Also compliance improves a company’s reputation, which should be a very potent ingredient for winning and maintaining contracts[1].

3. Increased “On-Time” Deliveries

One organization cited that they went from delivering products on time 51 percent of the time to 94 percent of the time. Some organizations have experienced a savings as high as 20 percent in their schedule[6]. “Generally, the more mature an organization is in the way it does business, the more successful it will be in delivering a quality product within project constraints.”[7]

4. Increased Quality

“[Participating] companies are looking at meeting their quality goals, meeting their requirements, building a maintainable product, and seeking better and improved quality as well as stabilizing schedule, meeting commitments, and accelerating or reducing schedule.”[6] Several software organizations have experienced a reduction in defects that ranged from as low as 10 percent to as high as 80 percent[6]. One organization reported a 45 percent decrease in its reduction error rate, while two more

organizations' product error rates decreased from 2.0 to 0.11 per thousand source lines of code and from 0.72 to 0.13 per thousand non-commented source statements[6].

Cons

1. Increased Spending for Process Implementation

In order for a software organization to mature, there has to be capital to support the effort. "Many organizations have expended large amounts of money and effort in support of their initiatives, yet they have little idea of what, if any, return they are accruing from their investments." [6] Costs for CMM-based process improvement programs have shown increases in software and hardware, data collection, design defects repair, code defects repair, first-time testing and overhead costs [6]. "One nearly universal complaint is that moving from level to level can cost hundreds of thousands or even millions of dollars." [1]

2. Increased Training Time, Decreased Manufacturing Time

Improving the software development process also includes increased training time and less time for working on projects. An organization has to continue business as usual or make the sacrifice to improve the process which will result in higher quality products. "Some organizations had difficulty finding the time to work on software-process improvement because they had extreme commitments to deliver customer products." [8] Additionally, in order to mature from one level to the next, extensive training is required so that the organization understands the processes.

PRODUCTIVITY ADJUSTMENT RECOMMENDATION

The following quote accurately sums up the purpose for collecting the most accurate and reflective productivity data from a contractor: "Little is known [of] the impact of software engineering practices and processes. Although much is written about the topic in qualitative terms, little quantitative information is available. In many ways, the engineering process is an information 'black hole' - it draws in money and resources like a magnet but little data emerges." [6] Many contractors have reported higher productivities as a result of CMM-based process improvement programs, but none has provided the data to substantiate its claim of success in achieving higher productivities.

Since no quantitative data exists which details the "true" impacts of process improvement, NCCA recommends the following procedures be employed when developing a contractor specific effort estimate: 1) the analyst should request the data that supports the conclusions of the contractor's independent assessment, and develop an effort estimate based on the new data and 2) if the contractor does not have quantitative data to support its maturity level rating, then the analyst should use NCCA's effort estimating tools and make no adjustments.

REFERENCES

- [1] Saiedian, Hossein and Richard Kuzara, "SEI Capability Maturity Model's Impact on Contractors," IEEE Computer, January 1995, pp. 16-25.
- [2] Fad, Bruce E., "Adapting Computer Aided Parametric Estimating (CAPE) to Process Maturity Levels," Ninth International COCOMO Estimating Meeting, October 1994, pp. 1-19.
- [3] Daskalantonakis, Michael K., "Achieving Higher SEI Levels," CrossTalk, September 1994, pp. 12-18.
- [4] Humphrey, Watts S., Terry R. Snyder and Ronald R. Willis, "Software Process Improvement at Hughes Aircraft," IEEE Software, pp. 11-23.
- [5] Dion, Ray, "Process Improvement and the Corporate Balance Sheet," CrossTalk, February 1994, pp. 7-15.
- [6] Brodman, Judith G. and Donna L. Johnson, "Return on Investment from Software Process Improvement as Measured by U.S. Industry," CrossTalk, April 1996, pp. 23-29.
- [7] Budlong, Faye C. and Judi A. Peterson, "Software Metrics Capability Evaluation Methodology and Implementation," CrossTalk, January 1996, pp. 15-19.
- [8] Paulish, Daniel J. and Anita D. Carleton, "Case Studies of Software Process-Improvement Measurement," IEEE Computer, 1994, pp. 50-57.